COMPUTER ENGINEERING WORKSHOP

S.E. (CIS) OEL REPORT

Project Group ID:

NAME OF MEMBER #1 Syed Ahad Nadeem CS-068
NAME OF MEMBER #2 Ammar Tariq CS-132
NAME OF MEMBER #3 Hamza Sheikh CS-126

BATCH: 2023

Department of Computer and Information Systems Engineering

NED University of Engg. & Tech.,

Karachi-7527

0CONTENTS

CHAPTER 1

PROBLEM DESCRIPTION

Develop an integrated environmental monitoring system in C, incorporating key programming concepts and practical applications. The project involves interacting with a free API to fetch real-time environmental data and implementing functionalities for data handling, automation, and alerts. The main objectives include:

- Fetching real-time environmental data (e.g., temperature,humidity) via a free API.
- Storing both raw and processed data in files for record-keeping.
- Creating shell scripts to automate tasks such as data retrieval andprocessing.
- Utilizing pointers and dynamic memory allocation to optimizedata handling and improve efficiency within the C program.
- Implementing real-time alerts using Linux system calls to notifypersonnel of critical environmental readings.
- Modularizing the code by using header files to improvereadability and maintainability.

2. Methodology:

We use C language to develop a functional automated weather reporting program. For fetching the weather details we have used a free weather API and coded the interface in Ubuntu Linux. The main purpose of using Linux is implementing automated notifications every **10 minutes** providing updated weather reports fetched by the API.

The libraries used area following:

`<stdio.h>`, `<stdlib.h>`, `<stdbool.h>`, `<string.h>curl/`

`curl.h and cJSON.`

Files

Source code:

The main.c file is the core of the program which provides the main logic to process and fetch data using libcurl and cJSON.

The monitoring_systems.h serves for declarations and as a helper in simple terminology.

Build and Execution:

The makefile obviously compiles the code and run_monitor.sh provides the shell script for running the system.

Data processing:

The raw_data.json inputs data for processing and processed_data.csv provides the processed output

Working

We use a free API called **OPENWEATHERMAP** as a source of weather data.

The system is built using a makefile , ensuring modular compilation and linking.

Execution is automated through the run_monitor.sh script, which sets up necessary environment variables and runs the binary.

The program is connected to the API for extracting environmental data via HTTP using **LIBCURL.** Error handling techniques are also incorporated for bugs and predictable issues like server and connection errors.

This raw json data's is parsed with json, and the metrics are extracted and validated such as Temperature, humidity etc. It also performs computations to derive and display additional insights by itself.

The output processed data is stored in processed_data.csv for visualization and additionally the program logs errors or warnings to assist in debuggingand system monitoring.

3. Results:

Data Acquisition:

Successfully retrieved environmental data from the API and multiple errors, mostly server errors, were encountered and handled appropriately

Processing:

Raw json data was parsed into a structured format.

The key metrics were extracted and additional outputs were computed

Outputs:

CSV file was organized data suitable for observation and visualization

Processed information was displayed accordingly

Overall the system proved to be effecient in handling real time data fetching and processing.